НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
Кафедра автоматизованих систем обробки інформації і управління

УДК: 004.422.81

«До захисту допущено»
В.о. завідувача кафедри
_____ *О.А.Павлов*
(підпис) (ініціали, прізвище)
“___” _____2019 р.

Дипломний проект
на здобуття ступеня бакалавра

з напряму підготовки *6.050101 «Комп’ютерні науки»*

на тему: «*Інформаційна система семантичного аналізу дописів у соціальних мережах*»

Виконав: студент *4* курсу, групи *ІС-51*

Гапанюк Георгій Дмитрович _____
(прізвище, ім’я, по батькові) (підпис)

Керівник *доц., к.т.н., доц. Баклан І.В.* _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант з графічної документації *доц., к.т.н., доц. Тєлишева Т.О.* _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент *доц. каф. ТК, к.т.н., доц. Ткач М.М.* _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цьому дипломному проекті немає запозичень з праць інших авторів без відповідних посилань.
Студент Гапанюк Г.Д. _____
(підпис)

Київ – 2019 року

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проекту складається з шести розділів, містить 4 рисунка, 14 таблиць, 1 додатку, 21 джерело.

Дипломний проект присвячений розробці комплексу цілей та задач з розробки інформаційної системи семантичного аналізу дописів в соціальних мережах, метою якою є оперативний семантичний аналіз великої кількості актуальної інформації яка з'являється в соціальних мережах.

Для розгляду загальних положень були описані діяльність системи, варіанти використання, виконаний огляд та аналіз існуючих аналогів та сформульовані задачі із визначеними цілями та метою.

Для опису інформаційного забезпечення були визначені вхідні та вихідні дані, описана база даних.

Для математичне забезпечення був описаний алгоритм, який використовується для семантичного аналізу дописів

Для визначення програмного та технічного забезпечення були описані основні засоби розробки, висунуті вимоги до технічного забезпечення, обрана та обґрунтована архітектура програмного забезпечення.

Була описана інструкція користувача та проведене тестування комплексу задач.

ДОПИС, ТЕМА, МОДЕЛЮВАННЯ, ОБРОБКА ПРИРОДНЬОЇ МОВИ, СОЦІАЛЬНА МЕРЕЖА

ABSTRACT

Structure and scope of work. The explanatory note of the diploma project consists of six sections, containing 7 figures, 5 tables, 1 supplement, 21 sources.

The diploma project is devoted to the development of a set of goals and objectives for the development of an information system for semantic analysis of posts in social networks, the purpose of which is an operational semantic analysis of a large number of actual information appearing in social networks.

For review of the general provisions, the activities of the system, usage patterns, review and analysis of existing analogues were described, and the goals with the defined goals and objectives were formulated.

In order to describe the information support, the input and output data were determined, the database was described.

For mathematical support, an algorithm used for semantic analysis of posts was described

To determine software and hardware, major development tools, technical support requirements have been described, software architecture has been selected and justified.

A user manual was described and a set of tasks was tested.

POST, TOPIC, MODELING, NATURAL LANGUAGE PROCESSING, SOCIAL NETWORK.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

API – Application Programming Interface (прикладний програмний інтерфейс);

ПЗ – програмне забезпечення;

ПК – персональний комп’ютер;

Тематичне моделювання – спосіб побудови моделі колекції текстових документів, яка визначає до яких тем відноситься кожен з документів

Семантичний аналіз – виділення семантичних відносин, формування семантичних уявлень тексту

Соціальна мережа – веб-сайт, який дозволяє користувачам створювати анкету та комунікувати з іншими користувачами

Тема – узагальнене коло явищ, які можливо охарактеризувати через ключові слова.

Документ – набір слів

Обробка природньої мови – напрям інформатики, що вивчає проблеми комп’ютерного аналізу та синтезу природної мови.

Стоп-слова – слова, які не несуть смислового навантаження, тому їх користь та роль для пошуку не суттєва.

Мішок слів – набір слів, що зустрічаються в документі без збереження їх розміщення в тексті.

Допис – текстове повідомлення, яке публікується користувачем в соціальних мережах.

ООП – об’єктно-орієнтоване програмування;

СУБД – система управління базами даних.

ВСТУП

З кожним днем частка людей, яка використовує інтернет для отримання інформації зростає. Наприклад, в Великій Британії частка людей у віці 18-24, що в основному отримують інформацію з Інтернету становить 84% [1]. Схожа ситуація й в Україні: Інтернет назвали основним джерелом інформації 72% респондентів у віці від 18-29 років [2]. Також неухильно росте частка людей, яка користується соціальними: в 2019 році кількість користувачів соцмереж зросла до 45% від населення землі, й з них 2,32 млрд користуються Facebook, а 1,9 млрд – YouTube [3]. Таким чином, соціальні мережі зараз є дуже потужними джерелами суспільної думки: їх користувачі читають та обговорюють новини в публічній площі, тим самим залишаючи по собі велику кількість даних, які не просто доступні для перегляду, але й можуть бути зібрані та оброблені сучасними інформаційними системами.

Ці дані можуть мати великий інтерес у людей, що займаються соціальною комунікацією: маркетологам, політтехнологам, вченим та соціологам. Наприклад, маркетологи можуть оцінювати та відстежувати охоплення маркетингових компаній, які вони проводять; політтехнологи можуть оцінювати, наскільки та чи інша новина викликала суспільний резонанс; соціологи можуть в динаміці відстежувати актуальні питання та рівень зацікавленості в них. Базуючись на цих даних, спеціалісти можуть робити аналіз суспільних настроїв та, наприклад, корегувати поточну інформаційну політику.

В інформатиці напрям, який займається подібною обробкою подібних даних, зветься «Обробкою природної мови» (англ. *Natural-language processing, NLP).* Швидкий розвиток інформаційних технологій надав потужний інструментарій для обробки людської мови, тим самим даючи можливість спеціалістам оброблювати надзвичайно великі масиви текстових даних й відокремлювати з них корисну інформацію, й чим більше люди будуть обговорювати соціально значимі питання, тим більшим буде попит на

програмне забезпечення, яке буде швидко збирати інформацію з соціальних мереж та надавати спеціалістам усі необхідні дані для аналізу. Отже, створення прикладних програмних систем для обробки природної мови є актуальним питанням.

З огляду на це метою даного проекту є розробка додатку для семантичного аналізу постів в соціальних мережах для покращеня відстежування динаміки суспільних настроїв в мережі.

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

Соціальні мережі – веб-сайт, який дає можливість користувачам створювати анкету з персональними даними та комунікувати з іншими користувачами через публічні та непублічні інструменти. В 2019 році соціальними мережами користується більше половини населення Землі й ця частка продовжує збільшуватися. Тим самим, це призводить до швидкого збільшення даних, які люди публікують про себе й які розміщені в публічному просторі. Сучасні соціальні мережі надають широкий спектр інструментів для збору та обробки такої інформації в комерційних та наукових цілях.

Розвиток інформаційних технологій сформував унікальний в історії період відкритості персональних даних, які можливо використати в широкому спектрі задачу. Тим самим, розвивається перспективний напрямок автоматизації процесів збору та семантичного аналізу подібних даних. Семантичний аналіз – це процес формування відношень між синтаксичними структурами та рівнем письма в цілому, до його незалежних від мови значень [4]. Одним з ключових різновидів семантичного аналізу є тематичне моделювання – спосіб побудови моделі колекцій текстових документів, яка визначає до який тем відноситься кожен з документів [5].

Даний дипломний проект націлений на автоматизацію процесу виділення теми серед постів в соціальних мережах, а саме – створення застосунку для завантаження та визначення теми серед великої кількості постів в соціальних мережах.

1.1.1 Опис процесу діяльності

Розглянемо процес діяльності системи, його описано за допомогою діаграми бізнес - процесу зображеної на рисунку 1.1.

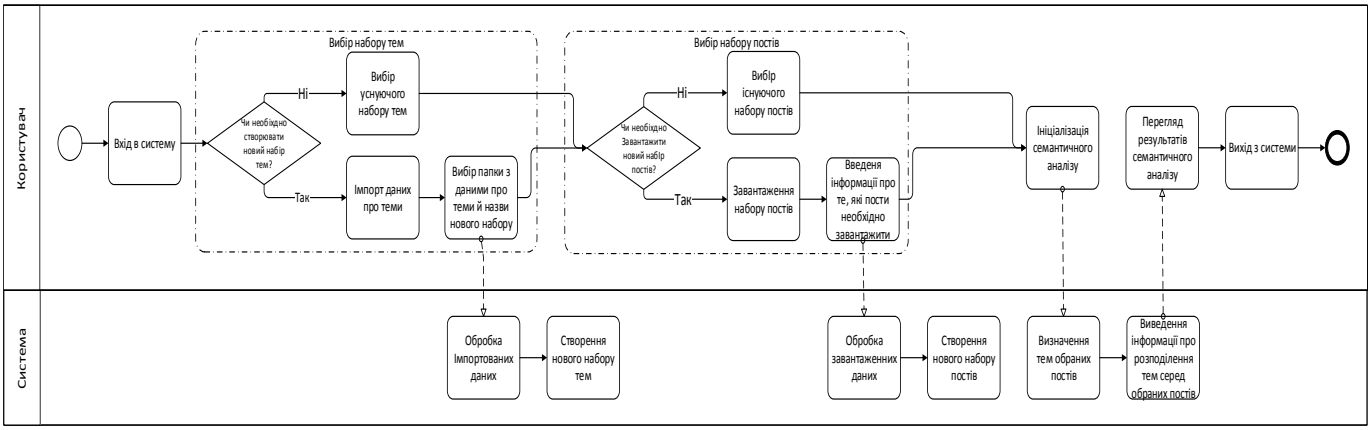

Рисунок 1.1 – Структурна схема бізнес-процесу

Після входу в систему необхідно обрати:

- набір постів, які будуть аналізуватися;
- набір тем, на приналежність до яких буде проходити аналіз.

Користувач може створити новий набір. Вхідна інформація для формування набору тем:

- назва теми;
- «мішок слів» (англ. Bag of words) цієї теми:
  1) слово, що зустрічається в темі;
  2) кількість входжень цього слова.

Вхідна інформація для формування набору постів:

- соціальна мережа;
- набір посилань;
- дати «з» та «по», за які відбираються дописи;
- додаткові «стоп-слова», які не будуть відібрані.

Після того, як користувач обирає або формує набори постів та тем, він може запустити семантичний аналіз. Після його виконання на екран виводиться сформована статистика.

Таким чином, користувачу стає доступні такі дані:

- розподілення за темами обраного набору постів;
- слова, характерні для обраних тем.

1.1.2 Опис функціональної моделі

Структурна схема використання наведена у графічному матеріалі.

Актор: Користувач

Варіанти використання

Таблиця 1.1– Типи залежностей між варіантами використання

Варіант використа-ння	*Опис дії варіанта використання*
Вибір набору тем	Користувач може обрати набір тем
Вибір набору дописів	Користувач може обрати набір дописів
Додавання набору тем	Користувач може імпортувати набір даних про теми й зберегти їх в системі
Додавання набору дописів	Користувач може імпортувати набір даних про дописи й зберегти їх в системі
Проведення семантичного аналізу	Користувач може запустити семантичний аналіз обраних дописів
Формування результатів семантичного аналізу	Користувач може сформувати результати семантичного аналізу
Видалення набору тем	Користувач може видалити набір тем
Видалення набору дописів	Користувач може видалити набір дописів

1.2 Огляд наявних аналогів

Інформація є одним з найважливіших ресурсів сучасного бізнес-середовища. Для будь-якої компанії важко досягти успіху, не маючи достатньої інформації про своїх клієнтів та співробітників.

Щодня компанії отримують неструктурований текст з різних джерел, таких як результати опитування, твіти, нотатки до колл-центру, телефонні розсилки, онлайн-відгуки клієнтів, записані взаємодії, електронні листи та інші документи. Й ці великі об’єми тексту нелегко обробити швидко без використання правильного інструменту аналізу тексту, яке допомагає користувачам отримувати інформацію з текстових даних, щоб діяти відповідно.

Аналіз тексту - це процес перетворення неструктурованих текстових даних у змістовні дані, які можливо використовувати для відстежування думки клієнтів та огляду продуктів. Програмне забезпечення для аналізу тексту використовує багато лінгвістичних, статистичних і машинних методів навчання.

1. GATE [6] – General Architecture for Text Enineering (*укр.* Загальна архітектура для текстової інженерії).Це набір інструментів, що використовуються для виконання різноманітних завдань з обробки природних мов. Розробка йшла в Університеті Шеффілда з 1995 року. Протягом багатьох років GATE виросла до клієнта на персональні комп’ютери, веб-додатка та бібліотеки Java. Програма включає в себе компоненти для вирішення різноманітних завдань з обробки мови.

   Переваги:

   – здатний вирішувати практично будь-яку проблему обробки тексту.
2. RapidMiner [7] є проектом з відкритим програмним кодом та широким спектром можливостей. Для RapidManer існує багато розширень, такі

як обробка тексту, «Weka», паралельна обробка, веб-майнінг, формування звітів й багато інших

Переваги:

- статистичний аналіз тексту;
- завантаження текстів з різних джерел даних;
- методи фільтрації та аналіз ваших текстових даних;
- підтримує кілька текстових форматів, включаючи звичайний текст, HTML або PDF, а також багато інших;
- стандартні фільтри для маркування, стемінгу, обробці стоп-слів.

3. KH Coder [8] - це безкоштовне програмне забезпечення для семантичного аналізу та дата-майнінгу. KH Coder підтримує японську, англійську, французьку, німецьку, італійську, португальську та іспанську мови. На вхід програми подається необроблений текст, який можливо детально аналізувати та зібрати інформативну статистику.

Переваги:

- обробка стоп-слів;
- список частоти використання слів;
- мережі зв’язку між словами;
- ієрархічний кластерний аналіз;
- можливість масштабування.

1.3 Постановка задачі

1.3.1 Призначення розробки

Суспільство переживає період масової інформатизації та все більше даних залишається в соціальних мережах. Новини та їх обговорення виникають в мережі моментально, а після Арабської Весни соціальні мережі стали й рушієм значних політичних змін, які розгортаються усього за дні. Це створює необхідність оперативно відслідковувати суспільні настрої для

політтехнологів, маркетологів, ньювсмейкерів та спецслужб та на їх основі приймати ті чи інші дії. Для аналізу поточних суспільних настроїв першочерговою являється відстежування тем, які зараз обговорюються в суспільному просторі.

1.3.2 Цілі та задачі розробки

Цілями розробки системи є створення інформаційної системи семантичного аналізу дописів в соціальних мережах, яка мала б наступні переваги:

- підтримка актуальності даних;
- можливість зміни тем та появлення нових;
- Інформативність щодо тем постів в соціальних мережах.

Для досягнення поставлених цілей необхідно реалізувати такі задачі.

Задачами розробки системи є:

- оперативне завантажування постів з соціальних мереж;
- формування власних наборів даних про теми, за якими буде проводитися аналіз постів;
- надання інформативної статистики щодо тем постів з соціальних мереж;
- компактність, легкість до розгортання та пристосування до зміни соціальних настроїв та лексики.

Із реалізацією задач буде створено компактна та портативна система, яка може бути використана для різноманітних ситуацій які потребують тематичного аналізу постів в соціальних мережах

Висновок до розділу

У даному розділі здійснений детальний аналіз предметної області, визначені бізнес-процеси системи. Оглянуто та проаналізовано існуючі аналоги. Визначено цілі задачі розробки.

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

Вхідні дані представлені у таблиці 2.1.

Таблиця 2.1 – Вхідні дані.

Данні	Опис
Дані про дописи	Дані про дописи, які завантажуються з соціальних мереж
Дані про теми	Дані про набір тем, за якими будуть розподілятися пости. Подаються на вхід в XML-форматі

2.2 Вихідні дані

Вихідні дані представлені в таблиці 2.2.

Таблиця 2.2 – Вихідні дані

Дані	Опис
Оброблені дані про теми	Дані про теми, які сформувалися після обробки поданих на вхід необроблених данів про теми. Зберігаються в БД
Дані про тематичний розподіл відібраних дописів	Результати семантичного аналізу даних, сформована статистика по дописам. Виводиться на екран після семантичного аналізу та зберігається в БД

2.3 Опис структури бази даних

В створеній базі даних SQL сформовані таблиці: «Posts», «Topics», «Result», «PostsResult»,.

Опишемо призначення кожної із них:

- Post - таблиця, яка містить інформацію про завантаженні дописи.
- Topic - таблиця, яка містить оброблену інформацію про теми;
- TopicSet – таблиця, яка містить інформацію про набори тем
- PostSet – таблиця, яка містить інформацію про набори дописів
- WordToTopics – таблиця, яка містить інформацію про те, наскільки слово відноситься до тієї чи іншої теми
- ResultLog – таблиця, яка містить інформацію про проведення семантичного аналізу

Структурна схема бази даних наведена у графічному матеріалі.

2.4 Структура масивів інформації

Для того, щоб подати на вхід дані про теми використовуються XML-файли. В таблиці подані теги.

Таблиця 2.3 – Теги XML-файлів

Тег	Опис
Document	Тег, який містить дані документа. Атрибути: Назва документа
Topic	Тег, який містить назву теми. Атрибути: вага теми в документі
Text	Тег, який містить слова документа. Атрибути: мова тексту

Висновок до розділу

У даному розділі було розглянуто вхідні та вихідні дані інформаційної системи. Описана структура бази даних.

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

Тематичне моделювання – це статистичне моделювання, яка використовується в машинному навчанні та обробці природньої мови. Воно характеризується виявленням абстрактних "тем", які фігурють в колекції документів. Тематичне моделювання є розповсюдженим інструментом text-mining'у для виявлення прихованих семантичних структур у тілі тексту. Інтуїтивно зрозуміло, що в документі, який стосується якоїсь теми частіше за інші зустрічаються окремі слова: "собака" і "кістка" частіше з'являться в документах про собак, "кіт" з'явиться в документах про кішок, але при цьому такі слова як «так» та «або» з'являться однаково в обох. Документ зазвичай стосується декількох тем у різних пропорціях; таким чином, у документі, який становить 10% про кішок і 90% про собак, напевно, буде приблизно в 9 разів більше слів про собак, ніж про котів. "Теми", сформульовані методами тематичного моделювання, є кластерами подібних слів. Тематичне моделювання фіксує цю інтуїцію в математичних рамках, які дозволяють досліджувати набір документів і виявляти, на основі статистичних даних щодо кожного, які теми можуть в них бути і яке розподілення тем для кожного документа.

Тематичне моделювання також називають імовірнісними моделюванням тем, які стосуються статистичних алгоритмів для виявлення прихованих семантичних структур великих обсягів тексту. У цифровому віці кількість текстових матеріалів, з якими ми стикаємося щодня, просто виходить за межі нашої спроможності до їх самостійного аналізу. Тематичне моделювання може допомогти організувати великий обсяг неструктурованих текстових даних. Спочатку розроблений як інструмент text-mining'у, тематичне моделювання можливо

використовувати для знаходження особливих структур в тексті, таких як: корисна інформація, зображення. Також його можливо використовувати в біоінформатиці [5].

3.2 Математична постановка задачі

Для тематичного моделювання буде використовуватися модель Латентного розподілу Діріхлє [9]. **M** показує кількість документві, **N** – набір слів з документа. Інші змінні перелічені нижче:

- α - параметр моделі, який показує розподіл тем по документу;
- β – параметр моделі, який показує розподіл слів по темам;
- θ_i - тематичний розподіл для документа *I*;
- φ_k – розподіл слів для теми *k*;
- z_{ij} – тема для *j*-го слова в документі *I;*
- ω_{ij} - слово.

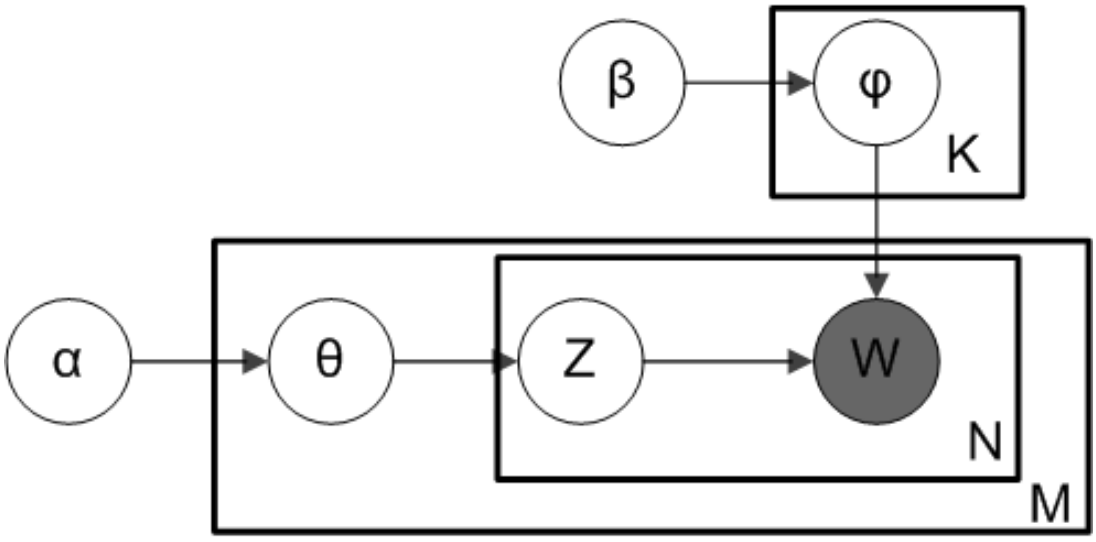

Рисунок 3.1 - Модель Латентного розподілення Діріхлє

Документ – це випадкове розподілення латентних тем, де кожна з тем характеризується розподіленням по усім словам. Латентний розподіл Діріхлє припускає наступний генеративний процес для корпусу D, що складається з M документів кожної довжини N_i:

- обрати θ_i ~ Dir(a), де i ∈ {1, …, M) та Dir(a) – розподіл Діріхлє з симетричним параметром α який зазвичай $a < 1$;
- обрати φ_k ~ Dir(β), де k ∈ {1, …, K) та Dir(β) – розподіл Діріхлє з симетричним параметром β який зазвичай $\beta < 1$;
- для кожного слова в точці i, j де i ∈ {1, …, M), та j ∈ {1, …, N_i}:
  1) обрати тему z_{ij} ~ Multinomial(θ_i);
  2) обрати слово ω_{ij} ~ Multinomial$(\varphi\ z_{ij})$.

Є набір з D документів. Кожен документ складається з N слів, ω_{ij} – слово в документі. Це всі змінні, які спостерігаються в моделі, інші змінні – приховані. Змінна z_{ij} приймає значення теми, обраної на кроці 2 для слова $\omega_{ij.}$ Для кожного документа d змінна θ_i – розподіл тем в цьому документі. В класичній моделі LDA кількість тем фіксована з початку й задається в явному виді параметром K. В моделі LDA змінні θ_i та φ_k розподілені так: θ_i ~ Dir(a), φ_k ~ Dir(β), де a та β – вектора-змінні розподілу Діріхлє. Як правило, всі компоненти параметрів a та β розподілення Діріхлє беруться однаковими, оскільки відсутня апріорна інформація щодо розподілення слів і тем в документах. Для того, щоб побудувати тематичну модель необхідно знайти приховані змінні.

3.3 Опис методів розв'язання

Генеративний процес LDA відповідає наступному спільному розподілу прихований та неприхованих змінних:

$$P(\boldsymbol{W}, \boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{\varphi}; \alpha, \beta) = \prod_{i=1}^{K} P(\varphi_i; \beta) \prod_{j=1}^{M} P(\theta_j; \alpha) \prod_{t=1}^{N} P(Z_{j,t} \mid \theta_j) P(W_{j,t} \mid \varphi_{Z_{j,t}}),$$

Рисунок 3.2 - Розподіл змінних генеративного процесу LDA

Для визначення оптимальних значень прихованих змінних моделі необхідно знайти так зване апостеріоне розподілення:

$$P(Z_{(m,n)} \mid \boldsymbol{Z}_{-(\boldsymbol{m},\boldsymbol{n})}, \boldsymbol{W}; \alpha, \beta) = \frac{P(Z_{(m,n)}, \boldsymbol{Z}_{-(\boldsymbol{m},\boldsymbol{n})}, \boldsymbol{W}; \alpha, \beta)}{P(\boldsymbol{Z}_{-(\boldsymbol{m},\boldsymbol{n})}, \boldsymbol{W}; \alpha, \beta)},$$

Рисунок 3.3 - Знаходження апостеріорного розподілення.

Висновок до розділу

В даному розділі дипломного проекту була математично описана модель, яка використовується для тематичного моделювання та детально описані алгоритми, які будуть використовуватися під час моделювання.

4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

Для розробки програмного забезпечення використовувалось:

- Операційна система **Windows 10** [10] – остання ос з Родини ОС «Microsoft Windows», що була випущена 29 липня 2015 року;
- База даних **SQLite** [11] **–** полегшена реляційна система керування базами даних. Втілена у вигляді бібліотеки, де реалізовано багато зі стандарту SQL-92;
- Середовище розробки **Visual Studio 2017** [12] **–** дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби**;**
- Мова програмування **C#** [13] **–** об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Розроблена Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде під егідою Microsoft Research (при фірмі Microsoft);
- **.NET Framework** [14] **–** програмна технологія, запропонована фірмою Microsoft як платформа для створення як звичайних програм, так і веб-застосунків. Багато в чому є продовженням ідей та принципів, покладених в технологію Java. Однією з ідей .NET є сумісність служб, написаних різними мовами;
- **Facebook SDK** [15] – набір засобів і документація для використання Facebook API при розробці програмних засобів на платформі .NET;

- **Google Api SDK** [16] - набір засобів і документація для використання Google API при розробці програмних засобів на платформі .NET;
- **LiveCharts** [17] – бібліотека засобів на платформі .NET, які дозволяють зручно візуалізувати дані у вигляді графів;
- **Microsoft Visio** [18] - редактор діаграм для Windows і редактор векторної графіки. Має широкий функціонал для створення UML-діаграм;
- **Microsoft Word** [19] – текстовий редактор, який дозволяє ефективно вести документацію;
- **ML.NET** [20] - це бібліотека, яка використовується для машинного навчання для програмних мов C #, F # і VB.NET. Також існує підтримка Python при використанні разом з NimbusML.

4.2 Вимоги до технічного забезпечення

4.2.1 Загальні вимоги

Застосунок має здійснювати роль інструмента, який дозволяє оцінювати тему великої кількості дописів в соціальних мережах за їх семантикою

Застосунок має виконувати наступні функції:

- вибір завчасно збереженого набору дописів;
- вибір завчасно збереженого набору даних про теми
- завантаження дописів з Інтернету;
- обробка тексту, чия тематична приналежність відома;
- тематичне моделювання дописів в соціальних мережах, чия тема невідома.

4.2.2 Вимоги до надійності

Програма повинна зберігати працездатність і забезпечувати відновлення своїх функцій при виникненні наступних позаштатних ситуацій:

- при втраті з’єднання з Інтернетом;
- при введені некоректних вхідних даних.

У разі виникнення аварійних ситуацій необхідно сповіщати користувача та надавати інструкцію для подальших дій.

4.2.3 Вимоги до складу і параметрів технічних засобів

Даний програмний продукт використовує дані з соціальних мереж, які завантажуються з мережі Інтернет, тому для коректної роботи програми необхідно мати стабільне підключення до Інтернету а також персональний комп'ютер з такими характеристиками:

- процесор з тактовою частотою не нижче 2 ГГц;
- об’єм оперативної пам’яті не менш ніж 1 Гб;
- операційні система Windows 10, Windows 8, Windows 7;
- встановлений .NET Framework 4.6;
- підключення до мережі інтернет.

4.3 Архітектура програмного забезпечення

4.3.1 Діаграма класів

Діаграма класів: статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення. Діаграма класів, також, може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм [21].

На рисунку 4.1 представлена діаграма класів. На ній зображена модель програмного забезпечення, заснована на класах та їх відношеннях. Ключові класи:

- Post – клас, який відноситься до допису в соціальній мережі. Містить властивості, які показують його айді в соціальній мережі, соціальну мережу, текст допису та дату публікації
- PostSet – набір даних про пости, які були завантажені в систему. Містить завантажені пости, назву набору та дату його завантаження.
- LDAScenario – клас сценарію, який реалізує тематичне моделювання на основі переданих наборів дописів та тем.
- TopicSet – набір даних про теми, які були завантажені в систему. Містить назву набору, дату його створення та коментар до нього.
- Topic – тема, яка містить назву теми та набори слів, які її характеризують
- Result – набір статичних даних щодо проведеного тематичного моделювання
- WordToTopics – відношення слів та їх «ваги» в якоїсь з тем

Структурна схема класів наведена у графічному матеріалі.

4.3.2 Діаграма діяльності

Діаграма діяльності - це візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій [21].

Діаграма на рисунку 4.2 описує процес діяльності користувача з системою. Користувач обирає або створює набір тем, обирає або створює набір постів, тим самим налаштовуючи систему для проведення

семантичного аналізу обраних даних. В кінці користувач ініціалізує семантичний аналіз та проводить аналіз створеної тематичної моделі.

Структурна схема діяльності наведена у графічному матеріалі.

4.3.3 Діаграма послідовності

Діаграма послідовності — це діаграма, яка відображає взаємодії об'єктів впорядкованих за часом. Зокрема, такі діаграми відображають задіяні об'єкти та послідовність відправлених повідомлень [21].

На рисунку 4.3. користувач спочатку обирає даних, на основі яких буде створений набір тем та обирає дані, за якими будуть завантажені та нормалізовані дописи з соціальних мереж. Після чого користувач обирає один з існуючих наборів тем та ініціалізує на основі обраних даних семантичний аналіз, за результатами якого будується тематична модель. На основі тематичної моделі формуються статистичні дані, які відображаються користувачу.

Структурна схема послідовності наведена у графічному матеріалі.

4.3.4 Діаграма компонентів

Діаграма компонентів — в UML, діаграма, на якій відображаються компоненти, залежності та зв'язки між ними.

Діаграма компонентів відображає залежності між компонентами програмного забезпечення, включаючи компоненти вихідних кодів, бінарні компоненти, та компоненти, що можуть виконуватись. Модуль програмного забезпечення може бути представлено як компоненту. Деякі компоненти існують під час компіляції, деякі — під час компонування, а деякі під час роботи програми [21].

Структурна схема компонентів наведена у графічному матеріалі.

4.3.5 Специфікація функцій

Дана інформаційна система використовує технологію WPF й основний функціонал розбитий на велику кількість методів й класів, які агрегуються в логічні компоненти у вигляді сценарій, тому в таблиці 4.1. виділені ключові сценарій та методи, які стосуються безпосередньо логіки системи

Таблиця 4.1 – Специфікація функцій

Сценарії	**Дія**
LDA	Сценарій, який формує тематичну модель
DownloadFacebookPosts	Сценарій, який завантажує з соціальної мережі Facebook дописи
DownloadYoutubePosts	Сценарій, який завантажує з соціальної мережі Youtube дописи
NormalizePosts	Сценарій, який нормалізує завантажені пости, тобто прибирає слова що не мають сенсу, прибирає суфікси та закінчення та рахує скільки раз нормалізоване слово зустрічається в наборі даних
NormalizePosts	Сценарій, який нормалізує завантажені пости, тобто прибирає слова що не мають сенсу, прибирає суфікси та закінчення та рахує скільки раз нормалізоване слово зустрічається в наборі даних
ImportTopicSets	Імпорт даних про теми з вказаного джерела

Продовження таблиці 4.1.

Сценарії	**Дія**
FormTopicData	Перетворення імпортованих даних в структурований вид, за якою буде характеризуватися тема
CreateResult	Створеня на основі тематичної моделі статистичних даних
ShowResult	Виведення статистичних даних, що були сформовані на основі тематичної моделі
SaveTopics	Збереження тем до бази даних
SavePosts	Збереження дописів до бази даних
DeleteTopicSet	Видалення збереженого набору тем
DeletePostSet	Видалення збереженого набору дописів

Висновок до розділу

В даному розділі були розглянути програмні продукти, які використовувалися при створені програмного продукту та супутньої документації, а також UML-діаграми. Були розглянуті вимоги до інформаційної системи та описані сценарії, які виконуються в створеному програмному продукті

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

При запуску програми користувачу відкривається головна сторінка інформаційної системи. Ця форма містить дані, які необхідно ввести перед початком семантичного аналізу:

- дані про теми;
- дані про дописи.

Дані про теми можуть бути вже збережені в базі даних або їх необхідно імпортувати. Для імпорту необхідно натиснути кнопку «Імпортувати теми», після чого користувачу відкриється діалог вибору файлів. Для того, щоб імпортувати теми необхідно обрати завчасно підготовлені XML-файли з даними про теми, після чого вони будуть проаналізовані системою та додані в базу даних. Після цього користувач може обрати теми, за якими він хотів би провести семантичний аналіз.

Дані про дописи можуть бути збережені в базі даних або їх необхідно завантажити з соціальних мереж. Для завантаження даних необхідно в розділі «Дописи» обрати соціальну мережу, обрати дати публікації дописів та ввести перелік посилань, за якими буде завантаженні дописи. Формат посилання залежить від соціальної мережі – для Facebook необхідно ввести ідентифікатор спільноти, для Youtube – ідентифікатор відео. Після цього необхідно обрати «Зберегти як» та у вікні, що відкрилося вписати назву набору вивантажених дописів. Після цього дописи будуть завантажені та нормалізовані у доступний для обробки вигляд.

Після того як всі необхідні дані були завантажені їх необхідно обрати та натиснути кнопку «Почати семантичний аналіз». У випадку тем необхідно обрати один набір тем, для дописів можливо, утримуючи клавішу «Shift», обрати декілька наборів дописів.

Після того, як інформаційна система виконає семантичний аналіз та побудує тематичну модель з’явиться окреме вікно зі статистичними даними про побудовану модель, яка буде містити такі дані: розподіл по темам обраного набору дописів, найбільш вживані слова й їх тематична належність. В кінці, як буде проведений семантичний аналіз, вікно статистики може бути зачинене та робота з системою продовжена або припинена.

Сторінка для завантаження та вибору наборів дописів, вікно вибору файлу з набором тем, меню вибору назви набору дописів наведені у графічному матеріалі.

5.2 Випробування програмного продукту

5.2.1 Мета випробувань

Метою випробувань являється перевірка відповідності функцій комплексу задач проведення семантичного аналізу дописів в соціальних мережах вимогам технічного завдання.

5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

5.2.3 Результати випробувань

В результаті тестування був перевірений весь функціонал інформаційної системи й був сформований детальний опис кожного з випробувань.

Таблиця 5.1. – Перевірка імпорт набору тем з коректного XML-файлу

Мета тесту	**Перевірка функції «Імпорт набору тем»**
Початковий стан моделі	Відкрита головна сторінка продукту
Вхідні дані:	Коректний XML-файл з даними про набори тем
Схема проведення тесту:	Натиснути кнопку «Імпорт тем». Обрати XML-файл, які містить набір даних про теми
Очікуваний результат:	Імпортований набір тем з’являється в комбінованому списку й доступний для вибору. Назва набору співпадає з назвою XML-файлу
Стан моделі після проведення випробувань:	Імпортований набір тем з’являється в комбінованому списку й доступний для вибору. Назва набору співпадає з назвою XML-файлу

Таблиця 5.2. – Перевірка імпорт набору тем з некоректного файлу

Мета тесту	**Перевірка функції «Імпорт набору тем»**
Початковий стан моделі	Відкрита головна сторінка продукту
Вхідні дані:	Любий файл не xml-формату або з порушенним xml-форматуванням
Схема проведення тесту:	Натиснути кнопку «Імпорт тем». Обрати XML-файл, які містить набір даних про теми

Продовження таблиці 5.2.

Мета тесту	**Перевірка функції «Імпорт набору тем»**
Очікуваний результат:	Показана спливаюча підказка про неуспішну спробу імпорту набору постів, перехід на головну сторінку застосування
Стан моделі після проведення випробувань:	Показана спливаюча підказка про неуспішну спробу імпорту набору постів, перехід на головну сторінку застосування

Таблиця 5.3. – Перевірка завантаження дописів з соціальної мережі Facebook

Мета тесту	**Перевірка функції «Завантаження дописів з Facebook»**
Початковий стан моделі	Відкрита головна сторінка продукту
Вхідні дані:	Набір посилань на спільноти в Facebook у форматі, перелічені через крапку з комою
Схема проведення тесту:	Обрати з списку «Соціальні мережі» Facebook. Ввести набір посилань на спільноти в поле «Посилання». Натиснути кнопку «Завантажити». У спливаючому вікні ввести назву набору дописів та натиснути «ОК»
Очікуваний результат:	Завантажений набір дописів з'являється в списку «Збережені пости» у вигляді строчки з назвою, введеною користувачем
Стан моделі після проведення випробувань:	Завантажений набір дописів з'являється в списку «Збережені пости» у вигляді строчки з назвою, введеною користувачем

Таблиця 5.4. – Перевірка завантаження дописів з Youtube

Мета тесту	**Перевірка функції «Завантаження дописів з Youtube»**
Початковий стан моделі	Відкрита головна сторінка продукту
Вхідні дані:	Набір посилань на відео в Youtube, перелічені через крапку з комою
Схема проведення тесту:	Обрати з списку «Соціальні мережі» Youtube. Ввести набір посилань на відео в поле «Посилання». Натиснути кнопку «Завантажити». У спливаючому вікні ввести назву набору дописів та натиснути «ОК»
Очікуваний результат:	Завантажений набір дописів з'являється в списку «Збережені пости» у вигляді строчки з назвою, введеною користувачем
Стан моделі після проведення випробувань:	Завантажений набір дописів з'являється в списку «Збережені пости» у вигляді строчки з назвою, введеною користувачем

Таблиця 5.5. – Перевірка аналізу введеного списку посилань на помилки

Мета тесту	**Перевірка функції «Перевірка списку посилань»**
Початковий стан моделі	Відкрита головна сторінка продукту
Вхідні дані:	Випадковий текст або посилання на інші веб-сторінки

Продовження таблиці 5.5.

Мета тесту	**Перевірка функції «Перевірка списку посилань»**
Схема проведення тесту:	Обрати з списку «Соціальні мережі» будь який варіант. Ввести випадковий текст в поле «Посилання». Натиснути кнопку «Завантажити»
Очікуваний результат:	Показана спливаюча підказка про некоректний список посилань, перехід на головну сторінку застосування
Стан моделі після проведення випробувань:	Показана спливаюча підказка про некоректний список посилань, перехід на головну сторінку застосування

Таблиця 5.6. – Перевірка запуску семантичного аналізу даних

Мета тесту	**Перевірка функції «Семантичний аналіз даних»**
Початковий стан моделі	Відкрита головна сторінка продукту
Вхідні дані:	Завантежені набір тем та дописи в базу даних
Схема проведення тесту:	Обрати з списку «Набіри тем» тему. Обрати зі списку «Дописи» набір дописів. Натиснути кнопку «Почати семантичний аналіз»

Продовження таблиці 5.6.

Мета тесту	**Перевірка функції «Семантичний аналіз даних»**
Очікуваний результат:	Відкривається вікно зі статистичними даними щодо побудованої тематичної моделі
Стан моделі після проведення випробувань:	Відкривається вікно зі статистичними даними щодо побудованої тематичної моделі

Таблиця 5.7. – Перевірка запуску семантичного аналізу даних

Мета тесту	**Перевірка функції «Семантичний аналіз даних»**
Початковий стан моделі	Відкрита головна сторінка продукту
Вхідні дані:	Відсутні
Схема проведення тесту:	Не обираючи набір тем та / або дописи, натиснути кнопку «Почати семантичний аналіз»
Очікуваний результат:	Показана спливаюча підказка про те, що набір тем та / або дописи не обрані
Стан моделі після проведення випробувань:	Показана спливаюча підказка про те, що набір тем та / або дописи не обрані

Таблиця 5.8. – Перевірка видалення набору дописів

Мета тесту	**Перевірка функції «Видалення набору дописів»**
Початковий стан моделі	Відкрита головна сторінка продукту
Вхідні дані:	Набір постів завантажений в систему та відображається в списку наборів дописів
Схема проведення тесту:	Обрати набір дописів зі списку. Обрати кнопку «Видалити дописи»

Продовження таблиці 5.8.

Мета тесту	Перевірка функції «Видалення набору дописів»
Очікуваний результат:	Обраний набір дописів зникає з переліку
Стан моделі після проведення випробувань:	Обраний набір дописів зникає з переліку

Таблиця 5.9. – Перевірка видалення набору тем

Мета тесту	**Перевірка функції «Видалення набору тем»**
Початковий стан моделі	Відкрита головна сторінка продукту
Вхідні дані:	Набір постів завантажений в систему та відображається в списку наборів тем
Схема проведення тесту:	Обрати набір дописів зі списку. Обрати кнопку «Видалити теми»
Очікуваний результат:	Обраний набір тем зникає з переліку
Стан моделі після проведення випробувань:	Обраний набір тем зникає з переліку

Висновок до розділу

В даному розділі була сформована детальна користувацьку інструкція з використання інформаційної системи, яка містить в собі детальний опис функціоналу.

Було проведене тестування функціоналу системи із детальним поясненням до нього.

ЗАГАЛЬНІ ВИСНОВКИ

У даній роботі була детально досліджена тема «Інформаційна система семантичного аналізу дописів в соціальних мережах»:

- був здійснений детальний аналіз предметної області, визначені бізнес-процеси системи. Оглянуто та проаналізовано існуючі аналоги. Визначено цілі задачі розробки;
- були розглянуті вхідні та вихідні дані веб-застосування наведені в таблицях. Була описана база даних, таблиця та відношення між ними;
- була математично описана модель, яка використовується для тематичного моделювання та детально описані алгоритми, які будуть використовуватися під час моделювання;
- були розглянути програмні продукти, які використовувалися при створені програмного продукту та супутньої документації, а також UML-діаграми. Були розглянуті вимоги до інформаційної системи та описані сценарії, які віконуються в створеному програмному продукті;
- була сформована детальна користувацька інструкція з використання інформаційної системи, яка містить в собі детальний опис функціоналу;
- було проведене тестування функціоналу системи із детальним поясненням до нього.

ПЕРЕЛІК ПОСИЛАНЬ

1. Where do people get their news? *Oxford University :* веб-сайт https://medium.com/oxford-university/where-do-people-get-their-news-8e850a0dea03 (дата звернення: 20.04.2019)
2. СМИ в Украине: наиболее используемые источники информации. *Research & Branding Group :* веб-сайт http://rb.com.ua/blog/smi-v-ukraine-naibolee-ispolzuemye-istochniki-informacii/ (дата звернення: 20.04.2019)
3. Global social media research summary 2019. *Smart Insights :* веб-сайт https://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-media-research/ (дата звернення: 20.04.2019)
4. *Goddard, Cliff* (2013). Semantic Analysis: An Introduction (2nd ed.). New York: Oxford University Press. p. 17.
5. *Blei, David* (April 2012). "Probabilistic Topic Models". Communications of the ACM. 55 (4): 77–84. doi:10.1145/2133806.2133826.
6. *General Architecture For Text Engineering* : веб-сайт https://gate.ac.uk/
7. *Rapid Miner* : веб-сайт https://rapidminer.com/
8. *KhCoder* : веб-сайт http://khcoder.net
9. *Blei, David M.; Ng, Andrew Y.; Jordan, Michael I* (January 2003). Lafferty, John (ed.). "Latent Dirichlet Allocation". Journal of Machine Learning Research. 3 (4–5): pp. 993–1022. doi:10.1162/jmlr.2003.3.4-5.993.
10. *Microsoft Windows* : веб-сайт https://www.microsoft.com/uk-ua/windows
11. *SQLite* : веб-сайт https://www.sqlite.org/index.html

12. *Visual Studio* : веб-сайт
https://visualstudio.microsoft.com/ru/
13. *C#* : веб-сайт
https://docs.microsoft.com/ru-ru/dotnet/csharp/
14. *.NET-Framework* : веб-сайт
https://dotnet.microsoft.com/
15. *Facebook API .NET SDK* : веб-сайт
https://github.com/facebook-csharp-sdk/facebook-csharp-sdk
16. *Google API .NET SDK* : веб-сайт
https://developers.google.com/api-client-library/dotnet/?hl=uk
17. *LiveCharts* : веб-сайт
https://lvcharts.net/
18. *Visio* : веб-сайт
https://products.office.com/uk-ua/visio/flowchart-software
19. *Microsoft Office* : веб-сайт
https://products.office.com/ru-ru/get-started-with-office-2019
20. *ML.NET* : веб-сайт
https://dotnet.microsoft.com/apps/machinelearning-ai/ml-dotnet
21. *James Rumbaugh, Ivar Jacobson, Grady Booch* (1999). The unified modeling language reference manual (англ.). Addison Wesley Longman Inc. ISBN 0-201-30998-X.

Додаток А

Тексти програмного коду

Інформаційна система семантичного аналізу дописів у соціальних мережах

(Найменування програми (документа))

DVD-RW

(Вид носія даних)

7 арк, 68654 кб

(Обсяг програми (документа) , арк.,)

Київ – 2019 року

```
<Window x:Class="SemanticAnalysis.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:SemanticAnalysis"
        mc:Ignorable="d"
        Title="Семантичний аналіз" Height="462.8" Width="640.8">
    <Grid Margin="0,0,-0.4,0.8">
        <Grid.RowDefinitions>
            <RowDefinition/>
            <RowDefinition Height="0*"/>
        </Grid.RowDefinitions>
        <ListBox x:Name="socials" HorizontalAlignment="Left" Height="57"
Margin="121,10,0,0" VerticalAlignment="Top" Width="191" ItemsSource="{Binding
Source={StaticResource enumValues}}" SelectionMode="Multiple"
SelectionChanged="Socials_SelectionChanged"/>
        <Label Content="Соціальні мережі" HorizontalAlignment="Left"
Margin="10,8,0,0" VerticalAlignment="Top" Height="26" Width="106"/>
        <Label Content="Дека з темами:" HorizontalAlignment="Left" Margin="10,75,0,0"
VerticalAlignment="Top" Height="26" Width="93"/>
        <TextBox x:Name="bagFolder" HorizontalAlignment="Left" Height="23"
Margin="121,78,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="191"/>
        <Label Content="Дата с" HorizontalAlignment="Left" Margin="10,114,0,0"
VerticalAlignment="Top" Height="25" Width="45"/>
        <DatePicker x:Name="dateFrom" HorizontalAlignment="Left" Margin="55,116,0,0"
VerticalAlignment="Top" Width="115" Height="24"/>
        <Label Content="по" HorizontalAlignment="Left" Margin="171,115,0,0"
VerticalAlignment="Top" Height="26" Width="24"/>
        <DatePicker x:Name="dateTo" HorizontalAlignment="Left" Margin="197,116,0,0"
VerticalAlignment="Top" Width="115" Height="24"/>
        <Label Content="Вхідні дані:" HorizontalAlignment="Left" Margin="10,144,0,0"
VerticalAlignment="Top" Width="93" Height="26"/>
        <TextBox x:Name="inputFile" HorizontalAlignment="Left" Height="23"
Margin="121,148,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="191"/>
        <Button x:Name="startButton" Content="Почати" HorizontalAlignment="Left"
Margin="214,185,0,0" VerticalAlignment="Top" Width="98" Height="30"
Click="StartButton_Click"/>
    </Grid>
</Window>
```

```
using SemanticAnalysis.Core;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace SemanticAnalysis
{
    /// <summary>
```

```
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public List<SocialNetworks> _socialNetworks;

        public MainWindow()
        {
            InitializeComponent();
        }

        private void Socials_SelectionChanged(object sender,
SelectionChangedEventArgs e)
        {
            _socialNetworks =
socials.SelectedItems.OfType<SocialNetworks>().ToList();
        }

        private void StartButton_Click(object sender, RoutedEventArgs e)
        {
            var scenarioParams = new LDAScenarioParams
            {
                BagOfWordsFolder = bagFolder.Text,
                DateFrom = dateFrom.SelectedDate.Value,
                DateTo = dateFrom.SelectedDate.Value,
                InputLinks = inputFile.Text,
                SocialNetworks = _socialNetworks
            };
            var scenario = new LDAScenario(scenarioParams);
            if (!scenario.Start())
            {
                MessageBox.Show(scenario.Errors);
            }
        }
    }
}


using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SemanticAnalysis.Core
{
    public class LDAScenario
    {
        LDAScenarioParams _params;
        public LDAScenario(LDAScenarioParams scenarioParams)
        {
            _params = scenarioParams;
        }

        public bool Start()
        {

            return true;
        }

        private string _errors;
        public string Errors { get { return _errors; } }
```

```
    }

    public class LDAScenarioParams
    {
        public List<SocialNetworks> SocialNetworks { get; set; }
        public string BagOfWordsFolder { get; set; }
        public DateTime DateFrom { get; set; }
        public DateTime DateTo { get; set; }
        public string InputLinks { get; set; }
    }

    public enum SocialNetworks
    {
        Facebook,
        YouTube
    }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
    <startup>
        <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6.1" />
    </startup>
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="Google.Apis" publicKeyToken="4b01fa6e34db77ab"
culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-1.40.0.0" newVersion="1.40.0.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="Google.Apis.Core" publicKeyToken="4b01fa6e34db77ab"
culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-1.40.0.0" newVersion="1.40.0.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="Newtonsoft.Json" publicKeyToken="30ad4fe6b2a6aeed"
culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-12.0.0.0" newVersion="12.0.0.0" />
      </dependentAssembly>
    </assemblyBinding>
  </runtime>
</configuration>
```

```
<Application x:Class="SemanticAnalysis.App"
             xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             xmlns:local="clr-namespace:SemanticAnalysis"
             StartupUri="View/MainWindow.xaml"
             xmlns:sys="clr-namespace:System;assembly=mscorlib"
             xmlns:core="clr-namespace:SemanticAnalysis.Core">
    <Application.Resources>
        <ObjectDataProvider x:Key="enumValues" MethodName="GetValues"
ObjectType="{x:Type sys:Enum}">
            <ObjectDataProvider.MethodParameters>
                <x:Type TypeName="core:SocialNetworks"/>
            </ObjectDataProvider.MethodParameters>
        </ObjectDataProvider>
    </Application.Resources>
</Application>
```

```
    public class Estimator
```

```
    {
        // output model
        protected Model trnModel;
        LDACommandLineOptions option;

        public bool init(LDACommandLineOptions option)
        {
            this.option = option;
            trnModel = new Model();

            trnModel.dfile = option.dfile;
            trnModel.dir = option.dir;
            trnModel.K = option.K;
            trnModel.savestep = option.savestep;
            trnModel.niters = option.niters;

            if (option.est)
            {
                if (!trnModel.initNewModel(option))
                    return false;
                trnModel.data.LocalDictionary.WriteWordMap(option.dir + "\\" +
option.wordMapFileName);
            }
            else if (option.estc)
            {
                if (!trnModel.initEstimatedModel(option))
                    return false;
            }

            return true;
        }


        public void estimate()
        {
            Console.WriteLine("Sampling " + trnModel.niters + " iteration!");

            int lastIter = trnModel.liter;
            for (trnModel.liter = lastIter + 1; trnModel.liter < trnModel.niters +
lastIter; trnModel.liter++)
            {
                Console.WriteLine("Iteration " + trnModel.liter + " ...");

                // for all z_i
                for (int m = 0; m < trnModel.M; m++)
                {
                    for (int n = 0; n < trnModel.data.Docs[m].Length; n++)
                    {
                        // z_i = z[m][n]
                        // sample from p(z_i|z_-i, w)
                        int topic = sampling(m, n);
                        if (topic < 50)
                        {
                            trnModel.z[m].Insert(n, topic);
                        }

                    }// end for each word
                }// end for each document

                if (option.savestep > 0)
                {
                    if (trnModel.liter % option.savestep == 0)
```

```
                    {
                        Console.WriteLine("Saving the model at iteration " +
trnModel.liter + " ...");
                        computeTheta();
                        computePhi();
                        trnModel.saveModel("model-" +
Conversion.ZeroPad(trnModel.liter, 5));
                    }
                }
            }// end iterations

            Console.WriteLine("Gibbs sampling completed!\n");
            Console.WriteLine("Saving the final model!\n");
            computeTheta();
            computePhi();
            trnModel.liter--;
            trnModel.saveModel("model-final");
        }

        /**
         * Do sampling
         * @param m document number
         * @param n word number
         * @return topic id
         */
        public int sampling(int m, int n)
        {
            // remove z_i from the count variable
            int topic = trnModel.z[m][n];
            int w = trnModel.data.Docs[m].Words[n];

            //initialize random number generator
            var rnd = new Random();

            trnModel.nw[w][topic] -= 1;
            trnModel.nd[m][topic] -= 1;
            trnModel.nwsum[topic] -= 1;
            trnModel.ndsum[m] -= 1;

            double Vbeta = trnModel.V * trnModel.beta;
            double Kalpha = trnModel.K * trnModel.alpha;

            //do multinominal sampling via cumulative method
            for (int k = 0; k < trnModel.K; k++)
            {
                trnModel.p[k] = (trnModel.nw[w][k] + trnModel.beta) /
(trnModel.nwsum[k] + Vbeta) *
                        (trnModel.nd[m][k] + trnModel.alpha) / (trnModel.ndsum[m] +
Kalpha);
            }

            // cumulate multinomial parameters
            for (int k = 1; k < trnModel.K; k++)
            {
                trnModel.p[k] += trnModel.p[k - 1];
            }

            // scaled sample because of unnormalized p[]
            double u = rnd.NextDouble() * trnModel.p[trnModel.K - 1];

            for (topic = 0; topic < trnModel.K; topic++)
            {
```

```
                if (trnModel.p[topic] > u) //sample topic w.r.t distribution p
                    break;
            }

            if (topic < 50)
            {
                trnModel.nw[w][topic] += 1;
                trnModel.nd[m][topic] += 1;
                trnModel.nwsum[topic] += 1;
                trnModel.ndsum[m] += 1;
            }

            return topic;
        }

        public void computeTheta()
        {
            for (int m = 0; m < trnModel.M; m++)
            {
                for (int k = 0; k < trnModel.K; k++)
                {
                    trnModel.theta[m][k] = (trnModel.nd[m][k] + trnModel.alpha) /
(trnModel.ndsum[m] + trnModel.K * trnModel.alpha);
                }
            }
        }

        public void computePhi()
        {
            for (int k = 0; k < trnModel.K; k++)
            {
                for (int w = 0; w < trnModel.V; w++)
                {
                    trnModel.phi[k][w] = (trnModel.nw[w][k] + trnModel.beta) /
(trnModel.nwsum[k] + trnModel.V * trnModel.beta);
                }
            }
        }
```